Deploying the Rubin Science Platform On Kubernetes & Openstack

Stelios Voutsinas
WFAU, Edinburgh

Rubin Science Platform

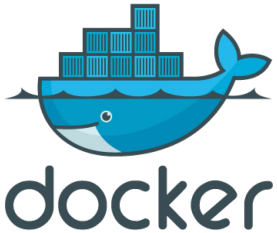
Rubin Platform consists of three aspects:

- Portal (Firefly)
- Notebook (Nublado)
- APIs (TAP, SODA)

Various services used for miscellaneous tasks. Some examples include:

- Gafaelfawr (Authentication)
- Cachemachine (Caching Docker Images)
- Moneypenny (Deploying Kubernetes Pods)
    - For example Deploying a user's Jupyter Environment
    - Every user get their own Kubernetes Namespace & pod

RSP - Technologies



RSP - Deployment Overview

- Deployed on Kubernetes, as a set of individual services (deployments in Kubernetes lingo)
- Kubernetes Cluster hosted on Openstack (Somerville), created using Magnum
  - Previously created via Rancher
- Every RSP service has one or more Docker images & which are packaged in a Helm chart
  - Upgrading a service means creating a new version of the Helm Chart & Docker Image
- Secrets stored in Vault
  - Vault stores secrets in Key/Value store
  - Examples of secrets include the ArgoCD user/pass, Database credentials, SSL certificates
  - Currently using the Rubin SQuare team Vault service, soon switching to one setup at ROE
  - RSP installation requires a *VAULT_TOKEN* for the Vault service to be used

RSP - Deployment Overview - ArgoCD

- The deployment of these Helm Charts is managed using ArgoCD
  - ArgoCD - GitOps tool, used for creating and managing Kubernetes deployments based on configuration stored in a Git repository
  - Deployed in the form of another Kubernetes application, it compares the current state against the Git repo state
  - Allows automated or guided upgrades/downgrades of Kubernetes deployments, via a single synchronization button
  - The ArgoCD CLI is used in the installation script, and the ArgoCD UI is used to administer and upgrade services.

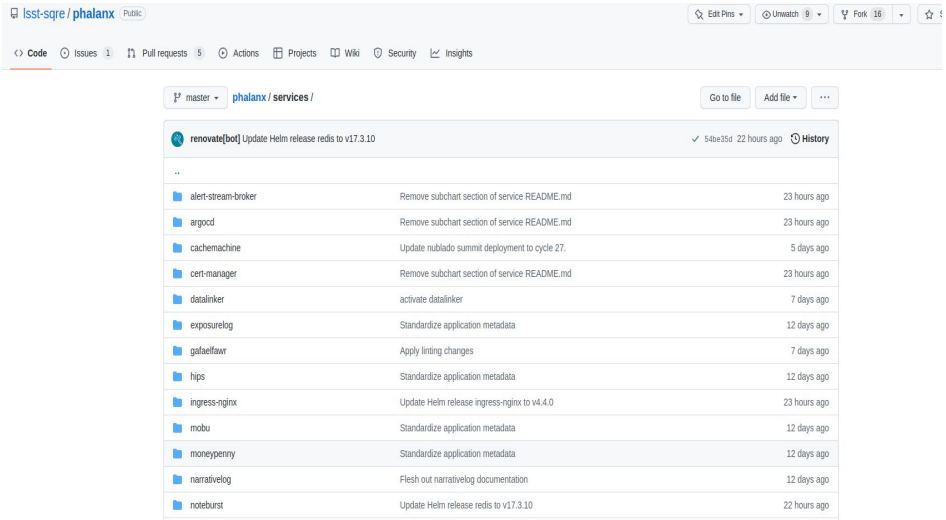
RSP - Deployment Overview - Storage

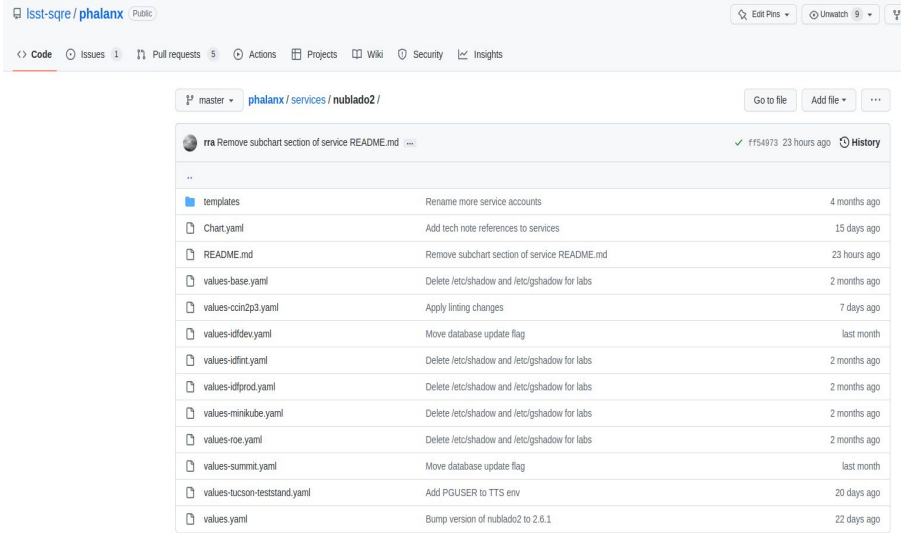
- NFS used for storing user & science data (images)
  - NFS mounted on each Nublado/Jupyter session, allowing persistent storage of notebooks & data
  - Science Data mounted on NFS VM stored in Openstack Ceph Volumes
  - Images accessible via the Butler tool, using Nublado
- The RSP also creates temporary Cinder volumes for Nublado/Jupyter user sessions
- QServ used for catalogue/science data, RSP provides access to it via APIs (TAP service) & the Portal (available soon)
- TAP query results written to Openstack Swift containers (via S3 API)

RSP - Deployment Overview - Phalanx

- Phalanx is the RSP's GitOps repository
  - Contains a configuration for each of the components
  - Configurations are environment specific (With some generic configs define for all environments)
  - ***ROE*** is our environment. When deploying we define this as the environment variable, and ArgoCD knows to pick our specific configurations
  - Our environment is included in the main repo, but we deploy from a forked repository. Gives us a bit more control of what is deployed here and gives us the chance to review changes when merging from the main repo.
  - Changes to our fork, get propagated when ready to main repository
- https://github.com/lsst-sqre/phalanx
- https://phalanx.lsst.io/

RSP - Deployment Overview - Phalanx





RSP - Deployment Walkthrough

- Containers & scripts available to make deployment as automated as possible
  - https://github.com/stvoutsin/phlx-installer
- Create Kubernetes Cluster

```
openstack coe cluster create --cluster-template workshop-template-piotr \
                             --master-count 1 \
                             --node-count 2 \
                             --docker-volume-size 200 \
                             --labels \
admission_control_list="NodeRestriction,NamespaceLifecycle,LimitRanger,ServiceAccount,ResourceQuota,TaintNodesBy
Condition,Priority,DefaultTolerationSeconds,DefaultStorageClass,StorageObjectInUseProtection,PersistentVolumeCla
imResize,MutatingAdmissionWebhook,ValidatingAdmissionWebhook,RuntimeClass" \
                             --merge-labels \
                             --keypair iris-rsp-1 \
                             stv-rsp-cluster
```

- Run script to open some additional ports (80, 443, 8443, 6379)

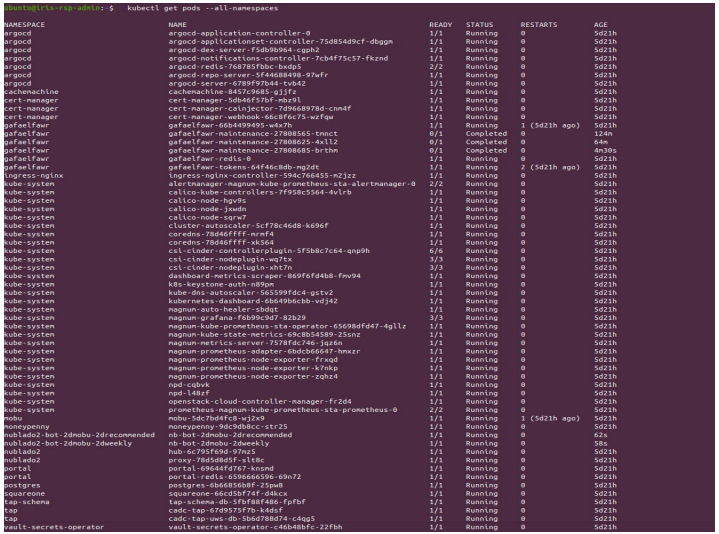
RSP - Deployment Walkthrough

- Copy kubernetes config & cluster certificate files to admin node
- Run docker container which handles the following:
  - Install required libraries for the installer
  - Create a standard Openstack storage class (Cinder) to be used for provisioning temp user volumes
  - Install the RSP

```
sudo docker run   \
  -it  \
  --hostname installer  \
  --env REPO=${REPO:?}  \
  --env VAULT_TOKEN=${VAULT_TOKEN:?}  \
  --env BRANCH=${BRANCH:?}  \
  --env ENVIRONMENT=${ENVIRONMENT:?}     \
  --volume ${CUR_DIRECTORY:?}"/phlx-installer/certs:/etc/kubernetes/certs"  \
  --volume ${CUR_DIRECTORY:?}"/phlx-installer/kube/config:/root/.kube/config" \
  --volume ${CUR_DIRECTORY:?}"/phlx-installer/scripts/install.sh:/root/install.sh"  \
  --volume ${CUR_DIRECTORY:?}"/phlx-installer/scripts/helper.sh:/root/helper.sh" \
  installer
```

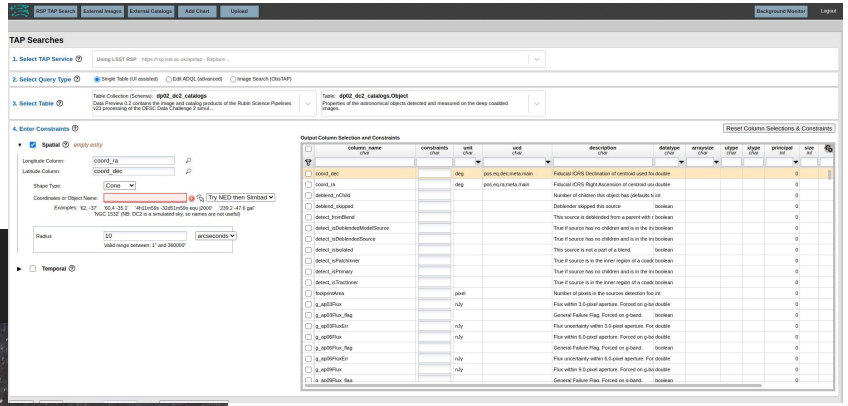
RSP - Deployment Validation

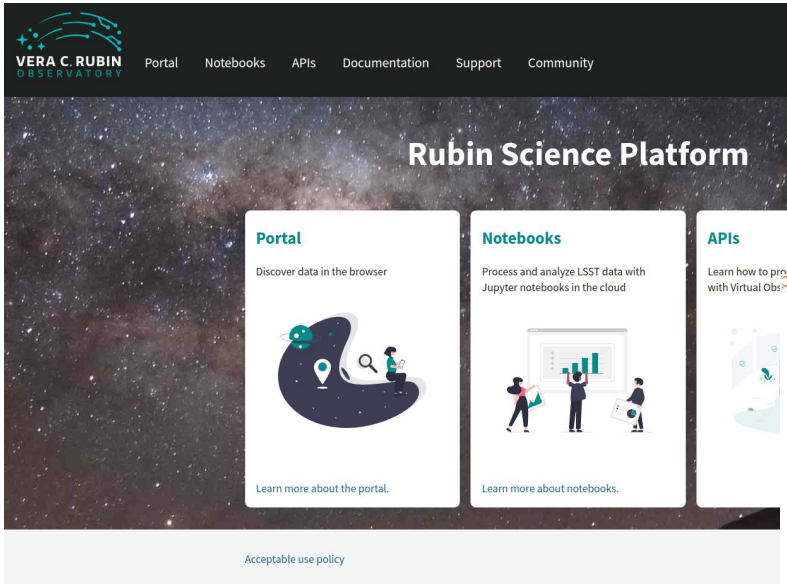
- `kubectl get pods --all-namespaces`

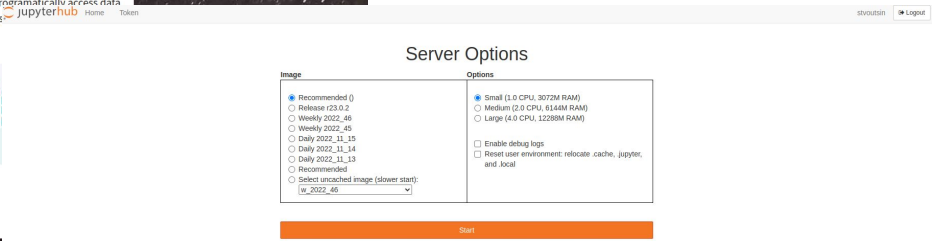


```
ubuntu@iris-rsp-admin:~$  kubectl get pods --all-namespaces
NAMESPACE                      NAME                                                      READY   STATUS      RESTARTS        AGE
argocd                         argocd-application-controller-0                           1/1     Running     0               5d21h
argocd                         argocd-applicationset-controller-75d854d9cf-dbggm         1/1     Running     0               5d21h
argocd                         argocd-dex-server-f5db9b964-cgph2                         1/1     Running     0               5d21h
argocd                         argocd-notifications-controller-7cb4f75c57-fkznd          1/1     Running     0               5d21h
argocd                         argocd-redis-76878Sfbbc-bxdpS                             2/2     Running     0               5d21h
argocd                         argocd-repo-server-5f446884988-97wfr                      1/1     Running     0               5d21h
argocd                         argocd-server-6789f97b44-tvb42                            1/1     Running     0               5d21h
cachemachine                   cachemachine-8457c9685-gjjfz                              1/1     Running     0               5d21h
cert-manager                   cert-manager-5db46f57bf-mbz9l                             1/1     Running     0               5d21h
cert-manager                   cert-manager-cainjector-7d9668978d-cnn4f                  1/1     Running     0               5d21h
cert-manager                   cert-manager-webhook-66c8f6c75-wzfqw                      1/1     Running     0               5d21h
gafaelfawr                     gafaelfawr-66b44994954-w4x7h                              1/1     Running     1 (5d21h ago)   5d21h
gafaelfawr                     gafaelfawr-maintenance-27808565-tnnct                     0/1     Completed   0               124m
gafaelfawr                     gafaelfawr-maintenance-27808625-4sll2                     0/1     Completed   0               64m
gafaelfawr                     gafaelfawr-maintenance-27808685-brthn                     0/1     Completed   0               4m30s
gafaelfawr                     gafaelfawr-redis-0                                        1/1     Running     0               5d21h
gafaelfawr                     gafaelfawr-tokens-64f46c8db-ng2dt                         1/1     Running     2 (5d21h ago)   5d21h
ingress-nginx                  ingress-nginx-controller-594c766455-m2jzz                 1/1     Running     0               5d21h
kube-system                    alertmanager-magnum-kube-prometheus-sta-alertmanager-0    2/2     Running     0               5d21h
kube-system                    calico-kube-controllers-7f958c5564-4vlrb                  1/1     Running     0               5d21h
kube-system                    calico-node-hgvds                                         1/1     Running     0               5d21h
kube-system                    calico-node-jxwdn                                         1/1     Running     0               5d21h
kube-system                    calico-node-sqrw7                                         1/1     Running     0               5d21h
kube-system                    cluster-autoscaler-5cf78c46d8-k696f                       1/1     Running     0               5d21h
kube-system                    coredns-78d46ffff-mrmf4                                   1/1     Running     0               5d21h
kube-system                    coredns-78d46ffff-xk564                                   1/1     Running     0               5d21h
kube-system                    csi-cinder-controllerplugin-5f5b8c7c64-qnp9h              6/6     Running     0               5d21h
kube-system                    csi-cinder-nodeplugin-wq7tx                               3/3     Running     0               5d21h
kube-system                    csi-cinder-nodeplugin-xht7n                               3/3     Running     0               5d21h
kube-system                    dashboard-metrics-scraper-869f6fd4b8-fmv94                1/1     Running     0               5d21h
kube-system                    k8s-keystone-auth-n89pm                                   1/1     Running     0               5d21h
kube-system                    kube-dns-autoscaler-565599fdc4-gstv2                      1/1     Running     0               5d21h
kube-system                    kubernetes-dashboard-6b649b6cbb-vdj42                     1/1     Running     0               5d21h
kube-system                    magnum-auto-healer-sbdqt                                  1/1     Running     0               5d21h
kube-system                    magnum-grafana-f6b99c9d7-82b29                            3/3     Running     0               5d21h
kube-system                    magnum-kube-prometheus-sta-operator-65698dfd47-4gllz      1/1     Running     0               5d21h
kube-system                    magnum-kube-state-metrics-69c8b54589-25snz                1/1     Running     0               5d21h
kube-system                    magnum-metrics-server-7578fdc746-jqz6n                    1/1     Running     0               5d21h
kube-system                    magnum-prometheus-adapter-6bdcb66647-hmxzr                1/1     Running     0               5d21h
kube-system                    magnum-prometheus-node-exporter-frxqd                     1/1     Running     0               5d21h
kube-system                    magnum-prometheus-node-exporter-kfnkp                     1/1     Running     0               5d21h
kube-system                    magnum-prometheus-node-exporter-zqhz4                     1/1     Running     0               5d21h
kube-system                    npd-cqbvk                                                 1/1     Running     0               5d21h
kube-system                    npd-l48zf                                                 1/1     Running     0               5d21h
kube-system                    openstack-cloud-controller-manager-fr2d4                  1/1     Running     0               5d21h
kube-system                    prometheus-magnum-kube-prometheus-sta-prometheus-0        2/2     Running     0               5d21h
nobu                           nobu-5dc7bd4fc8-wj2x9                                     1/1     Running     1 (5d21h ago)   5d21h
noneypenny                     noneypenny-9dc9db8cc-str25                                1/1     Running     0               5d21h
nublado2-bot-2dnobu-2drecommended nb-bot-2dnobu-2drecommended                            1/1     Running     0               62s
nublado2-bot-2dnobu-2dweekly   nb-bot-2dnobu-2dweekly                                    1/1     Running     0               58s
nublado2                       hub-6c7957f69d-97nz5                                      1/1     Running     0               5d21h
nublado2                       proxy-78d5d8d5f-slt8c                                     1/1     Running     0               5d21h
portal                         portal-69644fd767-knsnd                                   1/1     Running     0               5d21h
portal                         portal-redis-659666596-69n72                              1/1     Running     0               5d21h
postgres                       postgres-6b66856b8f-25pw8                                 1/1     Running     0               5d21h
squareone                      squareone-66cd5bf74f-d4kcx                                1/1     Running     0               5d21h
tap-schema                     tap-schema-db-5fbf88f486-fpfbf                            1/1     Running     0               5d21h
tap                            cadc-tap-67d9575f7b-k4dsf                                 1/1     Running     0               5d21h
tap                            cadc-tap-uws-db-5b6d788d74-c4qg5                          1/1     Running     0               5d21h
vault-secrets-operator         vault-secrets-operator-c46b48bfc-22fbh                    1/1     Running     0               5d21h
```

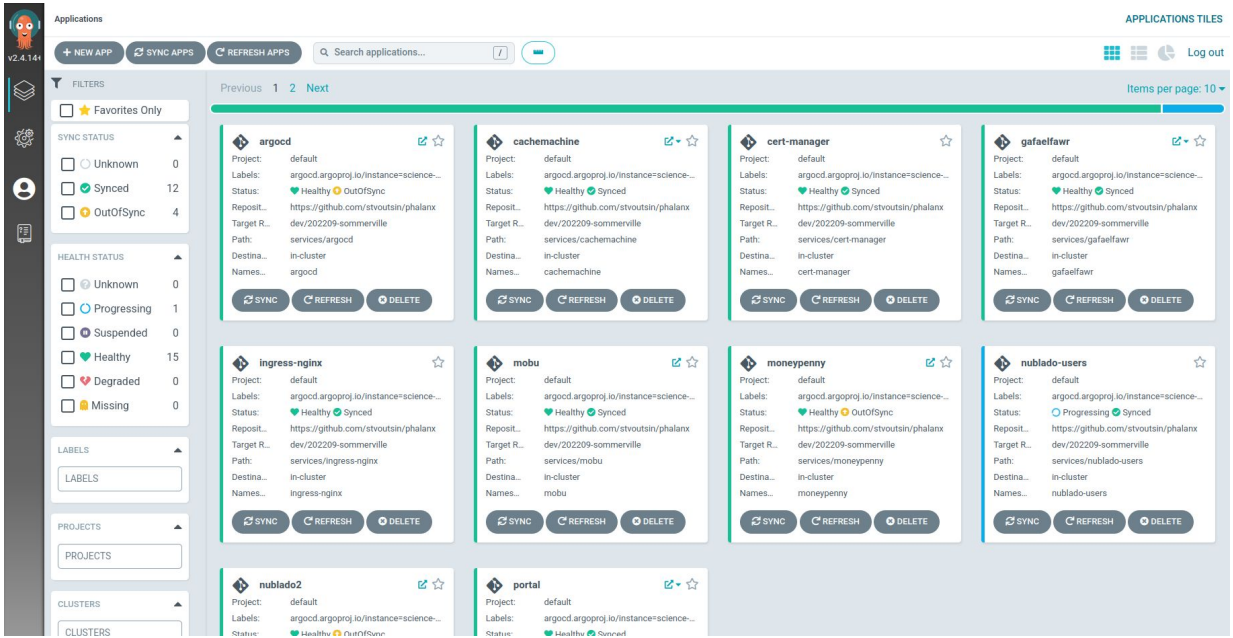
RSP - Deployment Validation



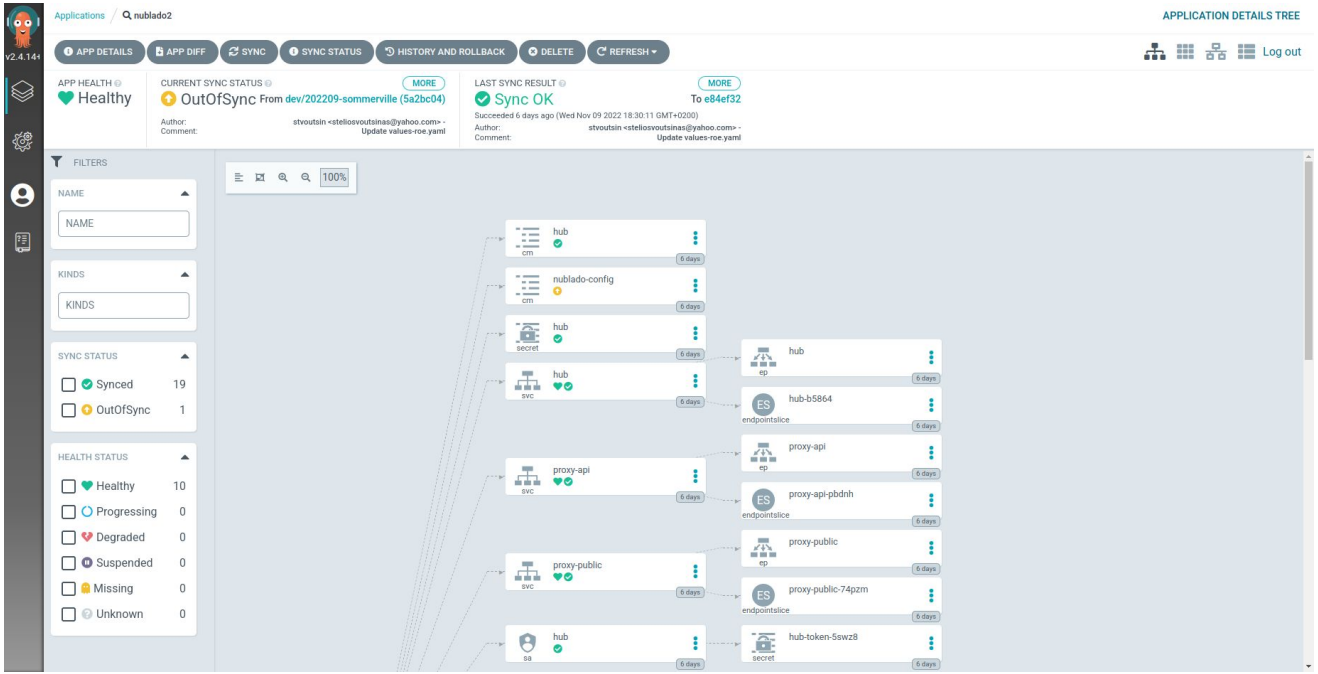




RSP - ArgoCD UI



RSP - ArgoCD UI – Upgrading Components



RSP - One-time Tasks

- Setup Vault Service & Vault secrets
  - Creating secrets can be done using a script included in the phalanx repo (`generate_secrets.py`)
- Create SSL certificate, add it to VAULT secrets.
  - Planning to move to Let's Encrypt, with a certbot that will allow automated cert renewal
- Create NFS VM & Ceph Volumes for User / Science Data
  - These are mounted on each user's Jupyter Server pod
- The installation could be a one time task, with upgrades being applied to the existing deployment.
  - We've tried two upgrade strategies:
    - Rebuild everything from scratch each time (Deploying nodes, Running install)
    - Upgrade components using ArgoCD without re-installation
  - Benefits and disadvantages to each, second option is what the SQuare team use, and is what we are doing as of recently

RSP - Advantages of RSP Design

- Self-serve nature of using Cloud technologies improves velocity of development
- For the most part, it is infrastructure agnostic
  - Should be able to easily transition between on-premise and cloud hosted infrastructure
- Upgrading components is painless
  - Except when things go wrong!
  - Single Synchronization step via ArgoCD
  - At Rubin (SQRE) this is a weekly task (Thursday Patch-day)
- Operations cost is minimized
- Continuous deployment becomes possible
- Self-healing built-in to the Kubernetes deployments
- Rollbacks are painless
- Deployments are transparent. Easier to get help from RSP development for specific environments

RSP - Issues / Difficulties

- Debugging problems with the RSP
  - Hard to track what the root causes are, due to complex architecture & technology stack
- Learning Curve & difficult bootstrap of new deployment / environment
- Dependence on Cloud Technologies
  - Rubin team use Google Cloud Services for various parts of the system
  - Google Cloud Storage was required until recently for the TAP service
- Phalanx & the RSP still in development, daily commits to master.
  - Merging with our forks gets tricky because this is a monthly tasks for us. I.e. Several hundred of changes to go through
  - Although usually merging everything in should be fine. Changes that break our setup are more rare, especially now that our environment exists in the main Phalanx repo

RSP - Issues / Difficulties specific to Openstack

- Kubernetes Networking & Openstack Security Policies
  - Magnum Pod Security Policy
  - Kubernetes CNIs (Calico) Issues when using Magnum with Service Type set to ClusterIP service type for the NGINX Ingress Controller
  - LoadBalancer Type worked, but not at first.
    - Issues with the Octavia OVN load balancer, need to ensure that at least one replica of the ingress controller pod is running on each of the nodes in the Octavia Load Balancer pool
    - It seemed that Octavia didn't automatically recognize which worker nodes have a replica so that only those are added to the pool. This may be fixed recently in our instance though
    - Healthchecks not yet available in our Openstack instance. Will be when OVN is upgraded to v22.09'
- Scaling Openstack Resources
  - Scaling resources requires a request and intervention from Openstack admins (i.e. if we need more Floating IPs, security groups or just more RAM, CPU etc..)

RSP - Load Balancing / Auto-Healing in Openstack & Kubernetes

- Magnum offers autoscaling and autohealing for worker nodes in container clusters.
- Users can specify the number of worker nodes needed and Magnum can automatically add or remove nodes based on the scaling policy.
- Magnum uses OpenStack Orchestration (Heat) to manage worker node lifecycle and supports autoscaling based on metrics like CPU or memory usage.
- Magnum provides self-healing for worker nodes by replacing unhealthy nodes with new instances to maintain cluster stability.
- Magnum's autoscaling and autohealing capabilities are limited to the infrastructure layer, not applications running in containers.
- The container orchestration engine used in Magnum clusters, such as Kubernetes, can provide application-level autoscaling and autohealing features.

RSP - Next Steps

- Use Let’s Encrypt for SSL Certificates
- Use our local Vault Service
- Experiment with various authentication systems (IAM?)
- Customize parts of the RSP (i.e. list of TAP services in Portal)

RSP - Links

- https://github.com/lsst-uk/rsp-uk-docs/wiki/RSP-Deployment-instructions-on-Openstack-with-Magnum
- https://github.com/stvoutsin/phlx-installer/tree/main/scripts
- https://phalanx.lsst.io/
- https://github.com/lsst-uk/phalanx
- https://github.com/lsst-sqre/phalanx